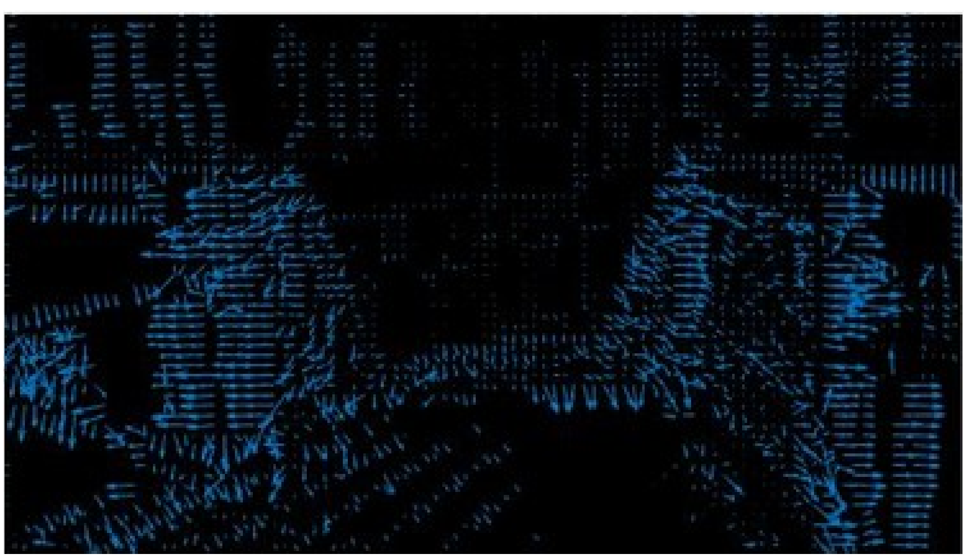
In the previous meeting, I intended to extract disparity using SIFT corner features and use their disparity to hypothesize on the depth of a certain region. Matching those SIFT features proved to be expensive and yielded sparse results. After some research I found that Lucas-Kanade's method for extracting optical flow is a much faster and more appropriate method. In paper [1] disparity was extracted using Lucas-Kanade (LK) method to hypothesize on the equations of the walls and ground-plane.

Lucas and Kanade [2] assumes brightness constancy and since the color of a pixel as the camera shifts stays the same the same pixel can be found in a consecutive frame by translating from it's location a certain amount. A hill-climber approach to finding the disparity of pixels is suggested in [2] to speed up the process.

The function estimateflow in MATLAB can use LK to return an optical flow image like that shown in Figure 1 . Apart form this method being faster it provided much more dense data compared to the SIFT method I had implemented which is necessary from hypothesizing at a more granular level.



In the past week I spent time experimenting with SIFT features before switching to using Lucas-Kanade optical flow. I learned that the video the disparity of features with motion was a high which cased noisy readings when matching small and detailed features. Resizing the images improved the results.

Having obtained seemingly usable data, I started thinking of how to hypothesize on depth, which I am currently still working on. To construct a hypothesis about a certain region in the image, the optical flow of features in that region have to be compared to some threshold optical flow. The threshold optical flow could be that which an image of an empty room (containing only a ground-plane and a far wall) would produce . If the flow observed was higher than that which we predict the empty room model would produce then the object which flow we are observing is closer than ground plane, otherwise if the flow was smaller then we could predict that there is a drop-off.

I am currently stuck figuring out how to predict that threshold flow. To facilitate the initial development I will assume that odometry is available and so I would not have to predict motion form the obtained optical flow. I will also initially assume that the camera is moving at a constant speed forward in the sample video I have.

After figuring out how to obtain an expected ground plane optical flow I intend to split the image into regions, propagate those regions with motion, update the certainty of the hypothesis as more flow vectors are observed in that region.